MKII

CONTEXT

CONTEXT

MKIV

Sometime in 2005 the development of LUATEX started, a further development of PDFTEX and a precursor to PDFTEX version 2. This TEX variant will provide:

- 21–32 bit internals plus a code cleanup
- flexible support for OpenType fonts
- an internal UTF data flow
- the bidirectional typesetting of ALEPH
- LUA callbacks to the most relevant TEX internals
- some extensions to TEX (for instance math)
- an efficient way to communicate with METAPOST

In the tradition of TEX this successor will be downward compatible in most essential parts and in the end, there is still PDFTEX version 1 as fall back.

In the mean time we have seen another unicode variant show up, XETEX which is under active development, uses external libraries, provides access to the fonts on the operating system, etc.

From the beginning, CONTEXT always worked with all engines. This was achieved by conditional code blocks: depending on what engine was used, different code was put in the format and/or used at runtime. Users normally were unaware of this. Examples of engines are ε-TEX, ALEPH, and XETEX. Because nowadays all engines provide the ε-TEX features, in August 2006 we started dropping standard TEX support in favour of ε-TEX features. This is a small effort because all code that is sensitive for optimization already has ε-TEX code branches for many years.

However, with the arrival of LUATEX, we need a more drastic approach. Quite some existing code can go away and will be replaced by different solutions. Where TEX code ends up in the format file, along with its state, LUA code will be initiated at run time, after a LUA instance is started. CONTEXT reserves its own instance of LUA.

Most of this will go unnoticed for the users because the user interface will not change. For developers however, we need to provide a mechanism to deal with these issues. This is why, for the first time in CONTEXT's history we will officially use a kind of version tag. When we changed the low level interface from Dutch to English we jokingly talked of version 2. So, it makes sense to follow this lead.

- **CONTEXT Mark I** At that moment we still had a low level Dutch interface, invisible for users but not for developers.
- **CONTEXT Mark II** We now have a low level English interface, which (as we indeed saw happen) triggers more development by users.
- **CONTEXT Mark IV** This is the next generation of CONTEXT, with parts reimplemented. It's an at some points drastic system overhaul.

Keep in mind that the functionality does not change, although in some places, for instance fonts, MkIV may provide additional functionality. The reason why most users will not notice the difference (maybe apart from performance and convenience) is that at the user interface level nothing changes (most of it deals with typesetting, not with low level details).

The hole in the numbering permits us to provide a Mark III version as well. Once XETEX is stable, we may use that slot for XETEX specific implementations.

As per August 2006 the banner is adapted to this distinction:

```
...  ver: 2006.09.06 22:46 MK II  fmt: 2006.9.6  ...
...  ver: 2006.09.06 22:47 MK IV  fmt: 2006.9.6  ...
```

This numbering system is reflected at the file level in such a way that we can keep developing the way we do, i.e. no files all over the place, in subdirectories, etc.

Most of the system's core files are not affected, but some may, like those dealing with fonts, input- and output encodings, file handling, etc. Those files may come with different suffixes:

- `somefile.tex`: the main file, implementing the interface and common code
- `somefile.mkii`: mostly existing code, suitable for good old TEX (ε-TEX, PDFTEX, ALEPH).
- `somefile.mkiv`: code optimized for use with LUATEX, which could follow completely different approaches
- `somefile.lua`: LUA code, loaded at format generation time and/or runtime

As said, some day `somefile.mkiii` code may show up. Which variant is loaded is determined automatically at format generation time as well as at run time.

In due time I may put some more text here, especially when we have a more clear picture of the components that make up MkIV.

Hans Hagen, Hasselt, August 2006^{++}